Amendments To The Claims:

This listing of claims will replace all prior versions and listings of claims in this application:

Listing of the Claims

1. (Currently Amended) A method ~~for a middle-tier server to impersonate~~ of impersonating a client to a plurality of servers, ~~the method~~ comprising:

obtaining by a middle tier server, a common nonce ~~associated with~~ that is created based at least in part upon a pre-nonce contribution from each of ~~the plurality of servers~~ a plurality of back-end servers, wherein the common nonce is generated from an entity other than the client that the middle tier server is to impersonate or the plurality of back-end servers that the middle tier server is to interact with on behalf of the client;

receiving by the middle tier server, a request from the client for a transaction with at least one of the plurality of back-end servers;

providing the common nonce from the middle tier server to the client;

receiving the common nonce signed by the client with the client's digital signature at the middle-tier server; and

impersonating the client by the middle tier server interacting with a selected one of the plurality of back-end servers for implementation of the client request on behalf of the client by providing the signed common nonce and the client request from the middle tier server to at least one of the plurality of back-end servers ~~as a signature for transactions~~ so as to authenticate the client to the plurality of servers for implementation of the client request on behalf of the client.

2. (Canceled)

3. (Currently Amended) The method of Claim 1 ~~Claim 2~~, wherein ~~the step of generating a common nonce comprises the steps of:~~ obtaining the common nonce comprises:

obtaining the pre-nonce contributions from the plurality of servers;

combining the pre-nonce contributions to provide a single pre-nonce token; and

providing the common nonce based on the pre-nonce token.

4. (Original) The method of Claim 3, wherein the step of providing the common nonce comprises reducing the pre-nonce token to provide the common nonce.

5. (Original) The method of Claim 3, wherein the step of combining the pre-nonce contributions to provide a single pre-nonce token comprises concatenating the pre-nonce contributions.

6. (Original) The method of Claim 4, wherein the step of reducing the pre-nonce token to provide the common nonce comprises the step of hashing the pre-nonce token utilizing a one-way hash function so as to provide the common nonce.

7. (Original) The method of Claim 3, wherein the step of obtaining pre-nonce contributions comprises the steps of:

requesting a pre-nonce contribution from each of the plurality of servers; and receiving the pre-nonce contributions from the plurality of servers.

8. (Original) The method of Claim 7, wherein requesting a pre-nonce contribution comprises sending authenticated requests to the plurality of servers.

9. (Original) The method of Claim 8, further comprising the step of encrypting the authenticated requests sent to the plurality of servers.

10. (Original) The method of Claim 8, wherein the authenticated requests include at least one of an identification of a source of the request, a time stamp and a random number.

11. (Original) The method of Claim 3, wherein the pre-nonce contributions include at least one of an identification of a server of the plurality of servers and a random number.

12. (Original) The method of Claim 3, wherein the pre-nonce contributions are signed with a signature corresponding to a server from which the pre-nonce contribution was obtained, the method further comprising incorporating the signatures in the pre-nonce token.

13. (Original) The method of Claim 3, wherein the pre-nonce contributions are signed with a signature corresponding to a server from which the pre-nonce contribution was obtained, the method further comprising authenticating the signatures of the pre-nonce contributions and rejecting pre-nonce contributions for which the digital signature is not authentic.

14. (Original) The method of Claim 3, further comprising the steps of:

receiving a transaction identification from a trusted server of the plurality of servers; and

associating the transaction identification with the common nonce.

15. (Original) The method of Claim 14, further comprising the step of tracking use of the common nonce based on the transaction identification.

16. (Original) The method of Claim 3, further comprising the steps of:

associating an expiration lime with a pre-nonce contribution; and

determining if the pre-nonce contribution has expired based on its associated expiration time.

17. (Original) The method of Claim 16, further comprising the steps of:

receiving the common nonce at a server of the plurality of servers;

determining a pre-nonce contribution associated with the received common nonce; and

accepting the received common nonce if the associated pre-nonce contribution has not expired.

18. (Original) The method of Claim 3, wherein at least one of the plurality of servers carries out the steps of:

receiving a client certificate;

determining if the client certificate is trusted; and

indicating that the client is not authenticated if the client certificate is not trusted.

19. (Original) The method of Claim 3, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce and a client certificate;

determining if the signature of the signed common nonce corresponds to a signature of the client certificate; and

indicating that the client is not authenticated if the signature of the signed common nonce does not correspond to the signature of the client certificate.

20. (Original) The method of Claim 6, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce, the common nonce and the pre-nonce token;

hashing the received pre-nonce token;

comparing the hashed pre-nonce token to the common nonce;

indicating that the client is not authenticated if the hashed pre-nonce token is different from the common nonce.

21. (Original) The method of Claim 11, wherein at least one of the plurality of servers carries out the steps of:

receiving the pre-nonce token;

determining if the pre-nonce token includes a random number associated with the at least one of the plurality of servers; and

indicating that the client is not authenticated if the pre-nonce token does not include the random number associated with the at least one of the plurality of servers.

22. (Original) The method of Claim 21, wherein at least one of the plurality of servers carries out the steps of:

associating an expiration with the random number associated with the at least one of the plurality of servers; and

indicating that the client is not authenticated if the pre-nonce token does not include a random number associated with the at least one of the plurality of servers which has not expired.

23. (Original) The method of Claim 1, wherein the step of obtaining a common nonce comprises the steps of:

obtaining the common nonce from a party trusted by the middle-tier server and the plurality of servers, the common nonce being signed by the trusted party; and verifying the signature of the common nonce is the signature of the trusted party.

24. (Original) The method of Claim 23, wherein at least one of the plurality of servers carries out the steps of:

receiving a client certificate;

determining if the client certificate is trusted; and

indicating that the client is not authenticated if the client certificate is not trusted.

25. (Original) The method of Claim 23, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce and a client certificate;

determining if the signature of the signed common nonce corresponds to a signature of the client certificate; and

indicating that the client is not authenticated if the signature of the signed common nonce does not correspond to the signature of the client certificate.

26. (Currently Amended) A system for ~~a middle-tier server to impersonate~~ impersonating a client to a plurality of servers, comprising:

means for obtaining by a middle tier server, a common nonce ~~associated with~~ that

is created based at least in part upon a pre-nonce contribution from each of ~~the plurality of servers~~ a plurality of back-end servers, wherein the common nonce is generated from an entity other than the client that the middle tier server is to impersonate or the plurality of back-end servers that the middle tier server is to interact with on behalf of the client;

means for receiving by the middle tier server, a request from the client for a transaction with at least one of the plurality of back-end servers;

means for providing the common nonce from the middle tier server to the client;

means for receiving the common nonce signed by the client with the client's digital signature at the middle-tier server; and

means for impersonating the client by the middle tier server interacting with a selected one of the plurality of back-end servers for implementation of the client request on behalf of the client by providing the signed common nonce and the client request from the middle tier server to at least one of the plurality of back-end servers ~~as a signature for transactions~~ so as to authenticate the client to the plurality of servers for implementation of the client request on behalf of the client.

27. (Currently Amended) A computer program product for ~~a middle-tier server to impersonate~~ impersonating a client to a plurality of servers, comprising:

a computer usable storage medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code that obtains by a middle tier server, a common nonce ~~associated with~~ that is created based at least in part upon a pre-nonce contribution from each of ~~the plurality of servers~~ a plurality of back-end servers, wherein the common nonce is generated from an entity other than the client that the middle tier server is to impersonate or the plurality of back-end servers that the middle tier server is to interact with on behalf of the client;

computer readable program code that receives by the middle tier server, a request from the client for a transaction with at least one of the plurality of back-end servers;

computer readable program code that provides the common nonce from the middle tier server to the client;

computer readable program code that receives the common nonce signed by the

client with the client's digital signature at the middle-tier server; and

computer readable program code that impersonates the client by the middle tier server interacting with a selected one of the plurality of back-end servers for implementation of the client request on behalf of the client by providing ~~provides~~ the signed common nonce and the client request from the middle tier server to at least one of the plurality of back-end servers ~~as a signature for transactions~~ so as to authenticate the client to the plurality of servers for implementation of the client request on behalf of the client.

28. (Currently Amended) A method of authenticating a client, comprising:

receiving a pre-nonce token and a common nonce that has been signed by a client at a back-end server of a plurality of back-end servers from a middle tier server that is impersonating the client, wherein:

the pre-nonce token comprises a combination of pre-nonce contributions from the plurality of back-end servers;

the common nonce is created by hashing the pre-nonce token and is generated from an entity other than the client that the middle tier server is impersonating or the plurality of back-end servers that the middle tier server is to interact with on behalf of the client; ~~a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; and~~

authenticating the client based on the received signed common ~~nonce.~~ nonce;

hashing the pre-nonce token using the same hashing technique used to create the common nonce from the pre-nonce token; and

verifying the middle tier server based upon a comparison of the received common nonce and hashed value of the received pre-nonce token.

29. (Original) The method of Claim 28, wherein the common nonce is provided by a trusted third party.

30. (Currently Amended) The method of Claim 28, wherein the common ~~notice~~ nonce is generated by the middle tier server. ~~an entity other than the client or the plurality of servers based on information provided by each of the plurality of servers.~~

31. (Currently Amended) A system for authenticating a client, comprising:

means for receiving a pre-nonce token and a common nonce that has been signed by a client at a back-end server of a plurality of back-end servers from a middle tier server that is impersonating the client, wherein:

the pre-nonce token comprises a combination of pre-nonce contributions from the plurality of back-end servers;

the common nonce is created by hashing the pre-nonce token and is generated from an entity other than the client that the middle tier server is impersonating or the plurality of back-end servers that the middle tier server is to interact with on behalf of the client; ~~a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; and~~

means for authenticating the client based on the received signed common ~~nonce.~~ nonce;

means for hashing the pre-nonce token using the same hashing technique used to create the common nonce from the pre-nonce token; and

means for verifying the middle tier server based upon a comparison of the received common nonce and hashed value of the received pre-nonce token.

32. (Previously Presented) A computer program product for authenticating a client, comprising:

a computer usable storage medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code which receives a pre-nonce token and a common nonce that has been signed by a client at a back-end server of a plurality of back-end servers from a middle tier server that is impersonating the client, wherein:

the pre-nonce token comprises a combination of pre-nonce contributions from the plurality of back-end servers;

the common nonce is created by hashing the pre-nonce token and is generated from an entity other than the client that the middle tier server is impersonating or the plurality of back-end servers that the middle tier server is to interact with on behalf of the client; ~~a common nonce that is provided to each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being associated with each of the plurality of servers and signed by the client; and~~

computer readable program code which authenticates the client based on the received signed common ~~nonce.~~ nonce;

computer readable program code which hashes the pre-nonce token using the same hashing technique used to create the common nonce from the pre-nonce token; and

computer readable program code which verifies the middle tier server based upon a comparison of the received common nonce and hashed value of the received pre-nonce token.

33. (New) The method according to claim 1, further comprising:

combining the pre-nonce contributions from the plurality of back-end servers into a pre-nonce token;

hashing the pre-nonce token by the middle-tier server to generate the common nonce; and

providing the pre-nonce token to the selected one of the plurality of back-end servers; wherein:

the selected back-end server hashes the pre-nonce token using the same hashing technique used by the middle-tier server and compares it to the verified common nonce thus authenticating both the client and the middle-tier server the selected back-end server.